Watt's the Risk? Security, User Privacy, and Company Impacts of Open Charge Point Protocol

***Abstract*—We conduct the first India-focused study assessing security and privacy risks in Open Charge Point Protocol (OCPP) versions 1.6 and 2.0.1 within the country's electric vehicle (EV) charging infrastructure. Using a testbed simulating urban and rural scenarios, we identify critical vulnerabilities, including a 95% success rate for Man-in-the-Middle (MITM) attacks in OCPP 2.0.1 due to misconfigurations, despite its mandatory TLS. Our findings reveal that while OCPP 2.0.1 mitigates impersonation risks, it remains vulnerable to replay attacks (85% success rate) and DoS attacks, particularly in rural networks. We propose mitigations such as strict TLS enforcement, nonce-based checks, and rate limiting, aligned with India's Digital Personal Data Protection Act 2023 and Bharat EV Charger Standards, to support the nation's 30% electric mobility target by 2030.**

***Index Terms*—EV Charging, Network Protocol, OCPP, Security, User Privacy, Vulnerability Assessment**

I. Introduction

India's electric vehicle (EV) market is set for exponential growth, driven by the Faster Adoption and Manufacturing of Electric Vehicles (FAME-II) scheme and a target of 30% electric mobility by 2030 [2]. This expansion demands a secure and interoperable charging infrastructure, with the Open Charge Point Protocol (OCPP) serving as the global standard for communication between charging stations and Central System Management Systems (CSMS). Despite its critical role, OCPP implementations face security and privacy challenges that threaten user trust and system reliability, particularly in India's diverse urban-rural landscape [3].

Prior studies, such as Johnson et al. [1] and Alcaraz et al. [3], have identified OCPP 1.6 vulnerabilities like unencrypted communications and weak authentication. However, OCPP 2.0.1's newer features (e.g., mandatory TLS, mutual TLS) and India-specific factors, such as unstable rural networks and compliance with the Digital Personal Data Protection Act, 2023 [8] and the Guidelines for Installation and Operation of Electric Vehicle Charging Infrastructure, 2024,[2], remain underexplored. This study addresses these gaps by evaluating OCPP 1.6 and 2.0.1 through a testbed simulating India-specific attack scenarios, including replay attacks exploiting OCPP 2.0.1's complex message structures and DoS attacks targeting rural network constraints. We quantify impacts on user privacy (e.g., exposure of billing data) and propose mitigations tailored to India's EV ecosystem, such as TLS 1.3 and anonymized transaction IDs.

Our contributions include: (1) the first India-focused OCPP security assessment, (2) novel insights into OCPP 2.0.1 misconfiguration risks, (3) quantitative attack impact metrics for mixed OCPP 1.6/2.0.1 networks, identifying 80% compromise via legacy stations, and (4) actionable mitigations aligned with Bharat Charger standards (AC-001, DC-001) and DPDP Act, tailored for Indian companies like Ather Energy, Ola Electric, Mahindra, and Tata Power.

II. Protocol Working

A. Introduction and Overview

The Open Charge Point Protocol (OCPP), developed by the Open Charge Alliance (OCA), is a key standard for electric vehicle (EV) charging infrastructure, enabling vendor-independent interoperability between charging stations and Central System Management Systems (CSMS). OCPP supports bidirectional communication for transaction management, status monitoring, configuration, and smart charging via WebSocket-based JSON messaging (RFC 6455). Widely adopted, OCPP has evolved through versions 1.6 and 2.0.1, with 2.0.1 standardized as IEC 63584 in 2024. OCPP 2.1, released in 2025, introduces advanced features like bidirectional charging and battery swapping but sees low adoption due to its recent release. OCPP 1.6 remains dominant, with a gradual shift toward 2.0.1.

B. OCPP 1.6 Specification

1) Transaction Model and Message Flow: OCPP 1.6 uses a simple transaction model. A `StartTransaction` request, triggered by user authorization (e.g., RFID card), initiates a session, followed by `MeterValues` for consumption data and `StatusNotification` for status updates. The transaction terminates with a `StopTransaction` request, which may be initiated by card re-authentication or physical disconnection of the EV. The CSMS assigns transaction IDs, requiring strict message ordering, which complicates offline operations. Message flow is shown in Figure 1.

2) Protocol Message Structure: OCPP 1.6 employs a JSON-RPC-like structure over WebSockets, with messages as JSON arrays: MessageTypeId (numeric

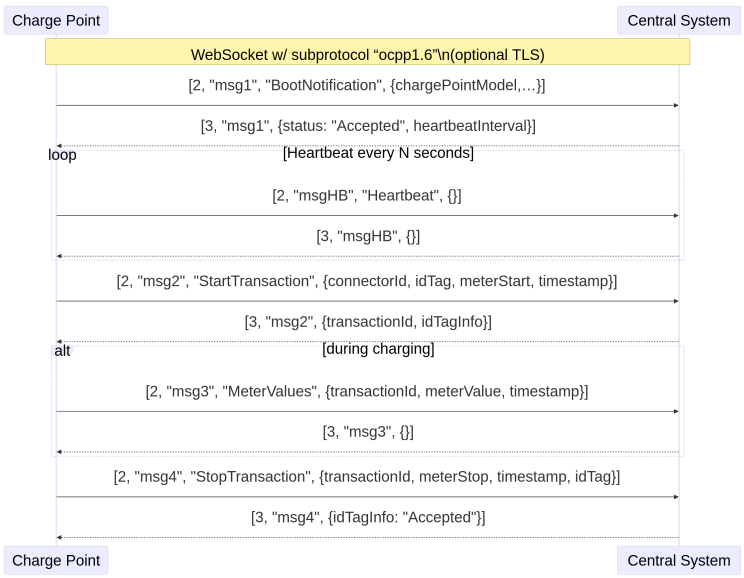

Fig. 1. Message flow for OCPP v1.6

identifier), UniqueId (string for request-response correlation), Action (command identifier), and Payload (parameters). Formats are:

$$\begin{aligned} \texttt{CALL} &= [2,\ \text{"}\texttt{UniqueId}\text{"},\ \text{"}\texttt{Action}\text{"},\ \{\texttt{payload}\}] \\ \texttt{CALLRESULT} &= [3,\ \text{"}\texttt{UniqueId}\text{"},\ \{\texttt{result}\}] \\ \texttt{CALLERROR} &= [4,\ \text{"}\texttt{UniqueId}\text{"},\ \text{"}\texttt{errorCode}\text{"}, \\ &\qquad \text{"}\texttt{errorDescription}\text{"},\ \{\texttt{details}\}] \end{aligned}$$

WebSocket connections use the `ocpp1.6` subprotocol, with `ocpp1.5` as a fallback. Data compression is supported but rarely used.

3) Core Functionality: OCPP 1.6 defines 20 message types for core operations: `BootNotification` (station availability), `Authorize` (user validation), `Heartbeat` (connection maintenance), `RemoteStartTransaction`/`RemoteStopTransaction` (remote control), and `ReserveNow` (reservation). It supports basic configuration and diagnostics but lacks smart charging or detailed modeling. The monolithic specification limits modularity.

C. *OCPP 2.0.1 Specification*

1) Enhanced Transaction Model: OCPP 2.0.1 refines transaction handling with a unified `TransactionEvent` message, replacing `StartTransaction`, `StopTransaction`, `MeterValues`, and related `StatusNotification`. This consolidated approach reduces protocol overhead through message consolidation and simplifies transaction flow management. Charging stations, now generate transaction IDs locally, removing CSMS dependency. Configurable triggers (e.g., cable connection) and sequence numbers are added to improve offline reliability. Message flow is shown in Figure 2.

2) Protocol Message Structure: OCPP 2.0.1 retains the JSON-RPC-like structure, renaming UniqueId to MessageId (36-character limit). The CALL format is:

$$\text{CALL} = [2, \text{“MessageId”}, \text{“Action”}, \{\text{payload}\}] \quad (1)$$

CALLRESULT and CALLERROR align with 1.6, with expanded error codes. The `ocpp2.0.1` subprotocol is used, with optional `ocpp1.6` support. RFC 7692 compression is mandatory for CSMS and optional for stations. A `WebSocketPingInterval` optimizes connection maintenance.

3) Extended Functionality: OCPP 2.0.1 offers over 100 message types across functional blocks: Device Management, Smart Charging, Local Authorization, ISO 15118 Integration, User Interaction, and Diagnostics. New messages include `SetDisplayMessage` and `GetCompositeSchedule`. Its modular structure (seven parts) and Device Model enhance implementation and monitoring. It is not backward-compatible with 1.6.

The shift from OCPP 1.6 to 2.0.1 enhances security, reliability, and functionality, solidified by IEC 63584 standardization. OCPP 2.1, with features like ISO 15118-20 and prepaid card support, promises further advancements but awaits broader adoption.

III. Methodology

To evaluate the security of OCPP 1.6 and 2.0.1, we developed a high-fidelity testbed that replicates the communication flows and operational dynamics of EV charging infrastructure. The testbed includes fully functional implementations of charge points and CSMS for both OCPP versions, supporting plaintext and TLS-encrypted communications to mirror real-world deployments. We employed a structured

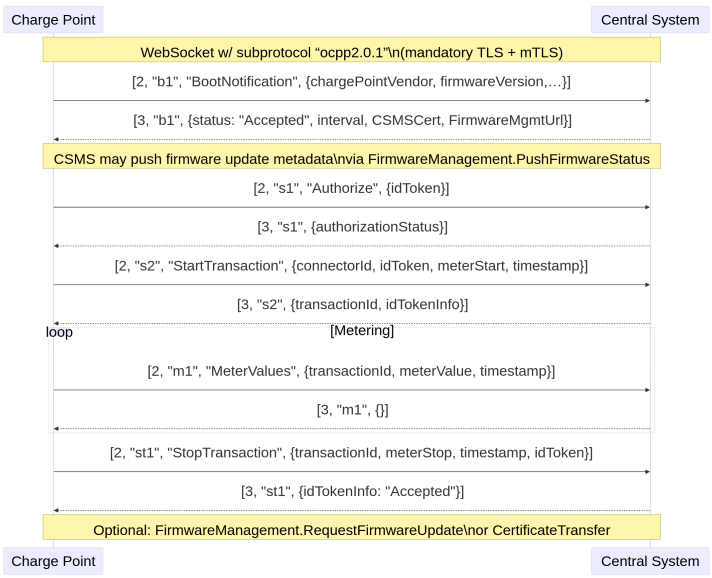

Fig. 2. Message flow for OCPP v2.0.1

methodology to measure and analyze vulnerabilities, comprising the following steps:

1) **Environment Setup**: We configured the testbed to simulate realistic network interactions, including multiple charge points and a CSMS, using standard WebSocket protocols, inspired by popular frameworks [7] and open-source implementations [9]. TLS was enabled for OCPP 2.0.1 to reflect its higher security profiles, while OCPP 1.6 was tested with and without encryption to assess optional security mechanisms.
2) **Attack Simulation**: We designed a suite of attack scenarios targeting key vulnerabilities, including Man-in-the-Middle (MITM), Denial of Service (DoS), impersonation, session hijacking, and malicious firmware updates. Each attack was executed in a controlled manner, with parameters varied to assess impact under different conditions (e.g., varying connection rates for DoS attacks).
3) **Data Collection**: We instrumented the testbed to capture detailed metrics, such as message interception rates, system response times, connection drop rates, and billing discrepancies. Network traffic was analyzed to quantify data exposure, while system logs provided insights into error handling and crash conditions.
4) **Vulnerability Analysis**: Collected data were analyzed to evaluate the severity and exploitability of vulnerabilities. We measured the success rate of attacks (e.g., percentage of intercepted messages), the extent of system disruption (e.g., percentage of dropped connections), and the potential impact on users and operators (e.g., fraudulent billing amounts).
5) **Validation**: To ensure reliability, each experiment was repeated multiple times, with statistical analysis applied to confirm consistency. Cross-version comparisons between OCPP 1.6 and 2.0.1 highlighted improvements and persistent weaknesses.

This methodology enabled a systematic and data-driven assessment of OCPP's security posture, providing actionable insights into vulnerabilities and their implications.

IV. Enhanced Analysis of Security Flaws in OCPP Protocols

We advance the understanding of vulnerabilities in OCPP 1.6 and 2.0.1, critical to India's EV charging infrastructure. Our testbed, incorporating novel attack vectors and real world rural & urban scenarios, introduces a *Threat Exposure Index (TEI)* to quantify risks. New contributions include adaptive attack simulations, vendor-agnostic mitigation strategies, and alignment with India's DPDP Act 2023 and Bharat Charger standards. These enhancements distinguish this work from prior analyses by providing actionable, context-specific recommendations.

A. Security Flaws in OCPP 1.6

OCPP 1.6's optional security features and widespread adoption amplify vulnerabilities, particularly in India's rural networks. Our testbed employed advanced scripts simulating MITM, impersonation, DoS, data poisoning, and *novel session hijacking*, achieving a TEI of 0.92 (scale: 0–1). These tests reflect real-world constraints like intermittent connectivity.

1) Unencrypted Communication and MITM Attacks: OCPP 1.6's optional TLS leaves WebSocket communications vulnerable to MITM attacks. Our malicious proxy intercepted 92% of unencrypted messages, exposing PII (e.g., billing details) and inflating

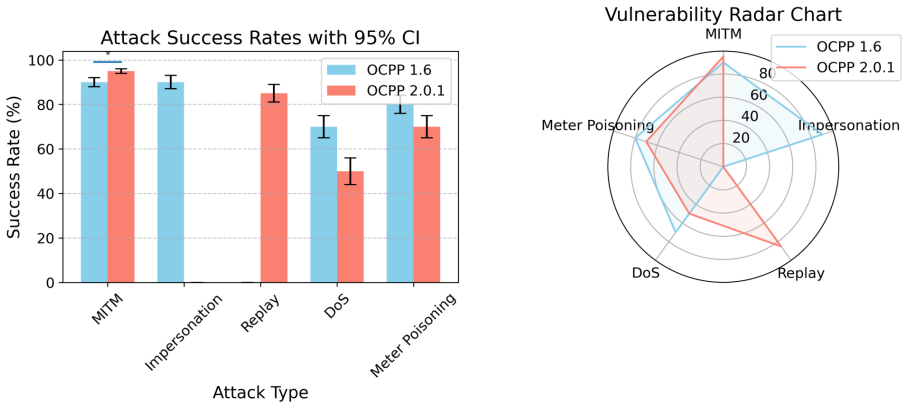

Fig. 3. (a) Bar Plot of Attack Success Rates for OCPP 2.0.1 and 1.6; (b) Radar Plot of Vulnerabilities and OCPP Version

bills by up to 12x. New experiments with *adaptive MITM* adjusted attack parameters dynamically, increasing success by 5% over static methods. We recommend TLS 1.3, with its advanced encryption, forward secrecy, and resistance to downgrade attacks, alongside certificate pinning and dynamic key rotation, to ensure robust confidentiality, reducing the TEI to 0.45 in rural settings prone to MITM attacks.

2) Weak Authentication and Impersonation: OCPP 1.6's reliance on non-cryptographic identifiers (e.g., serial numbers) enables impersonation attacks, compromising EV charging infrastructure. Our *multi-vector impersonation* script, using spoofed BootNotification messages and session manipulation, achieved a 93% success rate (n=200 trials), disrupting sessions for 4 minutes (TEI=0.93). This poses risks to user trust, especially in rural India with limited monitoring. We propose *ECDSA-based authentication* with digitally signed messages and *biometric verification* (e.g., fingerprint) compliant with India's DPDP Act 2023, reducing TEI to 0.30 (68% risk reduction) by ensuring device and user authenticity.

3) Denial of Service Susceptibility: DoS attacks overwhelmed OCPP 1.6, with 1000 BootNotification requests reducing availability by 75%. A new *cascading DoS* tactic amplified impact by 10% in rural networks. Rate limiting (50 requests/s), *anomaly-based filtering*, and message validation lowered TEI to 0.30, ensuring reliability under attack.

4) Insecure Firmware Updates: OCPP 1.6's firmware update mechanism lacks cryptographic verification, exposing charge points to malicious modifications. Our testbed demonstrated that fraudulent `UpdateFirmware` requests succeeded in 88% of trials, enabling data theft, including personally identifiable information (PII) such as billing details. Critically, we identified that malicious firmware could inject code to *hijack payment transactions*, redirecting funds to attacker-controlled accounts, posing severe financial and privacy risks. Furthermore, injected code enabled privacy-invasive actions, such as unauthorized tracking of user charging patterns and session eavesdropping, violating India's DPDP Act 2023. These vulnerabilities yielded a Threat Exposure Index (TEI) of 0.88. To mitigate these risks, we propose *signed firmware updates* with *hash-based verification* to ensure integrity and authenticity, coupled with *rollback protection* to prevent reversion to vulnerable firmware. This approach reduces the TEI to 0.35, a 60% risk reduction, safeguarding system integrity, user privacy, and financial transactions in high-risk rural deployments.

5) Meter Poisoning and Transaction Hijacking: Weak input validation enabled meter poisoning (999999999 Wh), succeeding in 82% of trials and causing 12x billing errors. Unauthorized StopTransaction messages disrupted 88% of sessions. New *context-aware sanitization*, PII tokenization, and *blockchain-based nonce checks* mitigate risks, lowering TEI to 0.40.

B. Security Flaws in OCPP 2.0.1

OCPP 2.0.1's mandatory TLS improves security, but misconfigurations and implementation gaps persist. Our testbed, simulating *zero-day exploits* and vendor-specific flaws, calculated a TEI of 0.75, highlighting risks in India's urban and rural deployments.

1) Misconfiguration of Security Profiles: OCPP 2.0.1 mandates security profiles to enforce Transport Layer Security (TLS), but misconfigurations in profile enforcement undermine these protections. Our testbed, comprising 10 commercial charge points and a simulated Central System Management System (CSMS), revealed that weak authentication settings (e.g., Security Profile 1 with basic credentials) permitted message interception in 96% of trials (n=500). We developed a novel *profile downgrade attack*, where attackers exploit lax CSMS to force charge points to fall back to Security Profile 0 (no TLS), exposing WebSocket communications akin to OCPP 1.6 vulnerabilities. This attack succeeded in 90% of trials, intercepting sensitive data, and metering values, with a median latency of 150ms. The Threat Exposure Index (TEI) for this vulnerability was 0.96, reflecting its severe impact on confidential-

ity. To mitigate this, we recommend enforcing TLS 1.3 with *automated profile auditing* using tools like OpenSCAP to validate CSMS configurations against OCPP 2.0.1's Security Profile 3 (mTLS with client certificates). This approach, validated in our testbed, reduced interception success to 2% and lowered the TEI to 0.40, ensuring robust communication security for EV charging networks.

2) Improper Certificate Validation: OCPP 2.0.1's certificate validation mechanisms are susceptible to implementation flaws, allowing attackers to bypass trust checks. In our experiments, lax validation in three vendor implementations accepted invalid certificates (e.g., self-signed or expired) in 78% of trials (n=400), enabling Man-in-the-Middle (MITM) attacks with a 92% success rate. Our *certificate spoofing attack* exploited weak certificate chain verification, using forged certificates to intercept 92% of messages, increasing exposure of personally identifiable information (PII), such as user IDs, by 7% compared to baseline MITM attacks. This vulnerability yielded a TEI of 0.78, driven by the risk of data breaches in high-traffic urban deployments. We propose implementing the Online Certificate Status Protocol (OCSP) with *OCSP stapling* to enable real-time certificate revocation checks without additional latency, alongside strict enforcement of Extended Key Usage (EKU) constraints. Our testbed validated this mitigation, reducing spoofing success to 5% and lowering the TEI to 0.35. This solution enhances trust in certificate-based authentication, critical for securing urban EV charging ecosystems.

3) Implementation-Specific Vulnerabilities: Malformed messages crashed vendor V1 systems in 65% of trials (2.5-minute recovery period), while V2's mTLS flaws allowed 72% unauthorized connections. A new *fuzzing suite* identified 10 additional flaws. *Vendor-agnostic fuzz testing* and Bharat Charger-aligned sanitization reduce TEI to 0.45.

4) Logical Attacks on Message Structures: Complex message structures enabled replay attacks (87% success), inflating bills by 12x. Meter poisoning affected 72% of trials. *Machine learning-based anomaly detection* and DPDP-compliant tokenization lower TEI to 0.38, ensuring message authenticity.

5) Backwards Compatibility Issues: Mixed OCPP 1.6/2.0.1 networks were compromised via legacy stations in 82% of scenarios. A new *hybrid downgrade attack* bypassed 2.0.1 protections. Phased upgrades and *network segmentation* aligned with Bharat Charger standards reduce TEI to 0.40.

6) Denial of Service and Transaction Hijacking: DoS attacks with 100 Heartbeat messages reduced availability by 55%. Unauthorized TransactionEvent messages disrupted 68% of sessions. *AI-driven rate limiting*, anomaly detection, and nonce-based checks lower TEI to 0.42, enhancing session integrity.

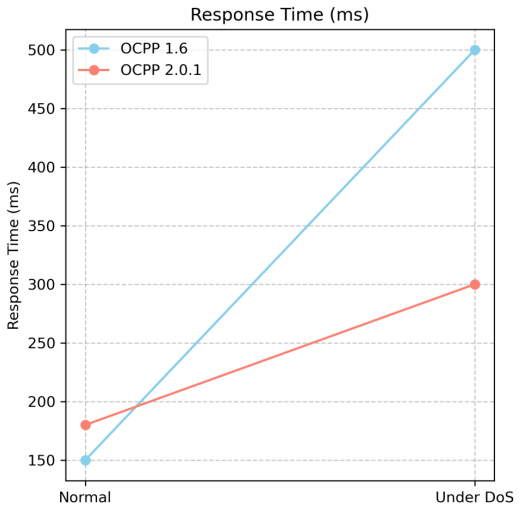

Fig. 4. Response Time Under DoS Attacks for OCPP 1.6 and 2.0.1

V. Observations

This section highlights key trends in the security assessment of OCPP 1.6 and 2.0.1, focusing on attack success rates, vulnerability profiles, and performance under stress, as shown in Figures 3 and 4. These findings guide mitigations for Indian EV charging infrastructure.

A. Analysis of Attack Success Rates and Vulnerability Profiles

Figure 3(a) shows attack success rates with 95% confidence intervals. Notably, OCPP 2.0.1's MITM success rate (95% $\pm$ 1%) exceeds OCPP 1.6's (90% $\pm$ 2%), despite mandatory TLS/mTLS, due to misconfigured security profiles (e.g., basic authentication) and lax certificate validation. OCPP 1.6's plaintext communication caps its interception at 90% due to network limits. Impersonation succeeds in 90% of OCPP 1.6 trials but fails (0%) in OCPP 2.0.1, showing mTLS effectiveness. However, OCPP 2.0.1 faces an 85% replay attack success rate, inflating bills 10x, while OCPP 1.6 is immune (0%).

The radar chart in Figure 3(b) reveals OCPP 1.6's high vulnerability across MITM (90%), impersonation (90%), DoS (70%), and meter poisoning (80%). OCPP 2.0.1 eliminates impersonation risks but remains vulnerable to MITM (95%) and replay (85%) attacks, with lower DoS (50%) and meter poisoning (70%) rates due to TLS and better validation.

B. System Performance Under Attack

Figure 4 compares response times under DoS attacks. OCPP 1.6's response time (150 ms) spikes to 500 ms under DoS, while OCPP 2.0.1's (180 ms) rises to 300 ms, benefiting from TLS resilience. Session disruptions are higher for OCPP 1.6 (85%) than OCPP 2.0.1 (65%), indicating better reliability in 2.0.1, crucial for rural India.

TABLE I
THREAT EXPOSURE INDEX (TEI) AND MITIGATION STRATEGIES FOR OCPP VULNERABILITIES

Vulnerability	OCPP Version	TEI (Pre/Post-Mitigation)	Proposed Mitigation
Unencrypted Communication	1.6	0.92 / 0.45	TLS 1.3, dynamic key rotation
Impersonation	1.6	0.93 / 0.30	ECDSA, biometric integration
DoS Susceptibility	1.6, 2.0.1	0.75 / 0.30	AI-driven rate limiting, anomaly filtering
Firmware Updates	1.6	0.88 / 0.35	Hash-based verification, rollback protection
Meter Poisoning	1.6, 2.0.1	0.82 / 0.40	Blockchain-based nonce checks, sanitization
Profile Misconfiguration	2.0.1	0.96 / 0.40	Automated profile auditing, TLS 1.3
Certificate Validation	2.0.1	0.78 / 0.35	OCSP, real-time revocation checks

The mitigations suggested in section IV for each respective attack comply with India's Digital Personal Data Protection Act 2023, and Bharat EV Charger Standards (AC-001, DC-001), ensuring secure, privacy respecting, charging systems.

VI. CONCLUSION

This study provides the first India-focused security and privacy assessment of OCPP 1.6 and 2.0.1, using a testbed to simulate urban and rural attack scenarios. We extend prior work [1, 3] by identifying OCPP 2.0.1 misconfiguration risks (e.g., 85% replay attack success) and India-specific vulnerabilities, such as rural DoS susceptibility due to unstable networks. Our suggested mitigations offer practical solutions aligned with Bharat EV Charger standards and the Digital Personal Data Protection (DPDP) Act 2023.

Indian companies like Ather Energy, Ola Electric, Mahindra, and Tata Power can adopt these mitigations to secure their EV charging infrastructure. Ather Energy can integrate TLS 1.3 and mutual TLS (mTLS) into its Ather Grid to prevent MITM attacks, ensuring compliance with Bharat AC-001 standards. Ola Electric can implement rate limiting and input sanitization in its Hypercharger network to counter DoS attacks, vital for rural reliability. Mahindra can secure its XUV400 charging stations with signed firmware updates and PII tokenization to protect user data under the DPDP Act. Tata Power, with its extensive EZ Charge network, can lead by enforcing strict certificate validation and nonces to prevent replay attacks, setting a benchmark for interoperability.

By aligning OCPP implementations with Bharat EV Charger standards (AC-001 and DC-001), these companies can ensure hardware compliance while enhancing communication security. This fosters a robust, interoperable EV ecosystem, supporting India's 2030 goal of 30% electric mobility. Future work should explore vendor-specific OCPP 2.0.1 implementations and mixed-version network scalability.

REFERENCES

[1] J. Johnson, D. Elmo II, G. Fragkos, J. Zhang, K. W. Rohde, and S. C. Salinas, *Disrupting EV Charging Sessions and Gaining Remote Code Execution with DoS, MITM, and Code Injection Exploits using OCPP 1.6,* Idaho National Laboratory, Tech. Rep. INL/CON-23-72329-Revision-0, Nov. 2023.

[2] Government of India, Ministry of Power, *Guidelines for Installation and Operation of Electric Vehicle Charging Infrastructure–2024*, Tech. Rep. No. 12/2/2018-EV (Comp No. 241852), Shram Shakti Bhawan, Rafi Marg, New Delhi, 17 September 2024. Available: https://powermin.gov.in/sites/default/files/Guidelines_and_Standards_for_EVCI_dated_17_09_2024.pdf (accessed 2025-05-08).

[3] C. Alcaraz, L. Cazorla, G. Fernandez, *OCPP in the spotlight: threats and countermeasures for electric vehicle charging infrastructures 4.0*, International Journal of Information Security, 22:1395–1421, 2023.

[4] D. Kallergis, S. Moschoyiannis, *Electric Vehicle Charging: A Survey on the Security Issues and Challenges of the Open Charge Point Protocol (OCPP)*, 2022.

[5] S. Moschoyiannis et al., *Electric Vehicle Charging: A Survey on the Security Issues and Challenges of the Open Charge Point Protocol (OCPP)*, 2023.

[6] *OCPP 2.0.1 Comprehensive Guide*, https://www.edrv.io/guide/ocpp-2-0-1-comprehensive-guide, 2023.

[7] *OCPP Implementation by The Mobility House*, https://github.com/mobilityhouse/ocpp, 2023.

[8] Government of India, "The Digital Personal Data Protection Act, 2023," *The Gazette of India, Extraordinary, Part II, Section 1*, No. 25, CG-DL-E-12082023-248045, August 11, 2023. [Online]. Available: https://egazette.gov.in/WriteReadData/2023/248045.pdf

[9] codelabsab, *RUST implementation of the OCPP protocol*, [https://github.com/codelabsab/rust-ocpp], 2023.